

Python 语法手册及编程练习

作者：王琦

创建日期：2017 年 5 月 10 日

最后修改：2017 年 5 月 10 日

摘要：本文提供了 Python 3 标准下的部分 Python 语法，每个知识点都由范例、参考链接和练习组成，读者可在阅读内容后完成相应练习以巩固理解。由于支持 Python 3 的 Online Judge 较少，在此我选择了 [HackerRank Online Judge](#) 作为练习网站。

0. 参考文献

电子书： [《Dive Into Python 3》](#) 和 [Python 官方文档](#)。

1. 列表 (Lists)

1.1 语法

[《Dive Into Python 3》2.4 Lists](#) 描述了列表的用法，并指出 列表不仅仅是数组：

Lists are Python's workhorse datatype. When I say "list," you might be thinking "array whose size I have to declare in advance, that can only contain items of the same type." Don't think that. Lists are much cooler than that.

- A list in Python is like an array in Perl 5. In Perl 5, variables that store arrays always start with the @ character; in Python, variables can be named anything, and Python keeps track of the datatype internally.
- A list in Python is much more than an array in Java (although it can be used as one if that's really all you want out of life). A better analogy would be to the ArrayList class, which can hold arbitrary objects and can expand dynamically as new items are added.

1.2 练习

请完成下述练习：

- 翻转数组： [HackerRank. Arrays](#)
- 处理二维数组： [HackerRank. 2D Arrays](#)
- 数组左移： [HackerRank. Left Rotation](#)

2. 字典 (Dictionaries)

2.1 语法

在 Python 3 中，字典是由键值对 (key-value pair) 组成的无序集 (unordered set)，它是使用哈希表 (hash table) 实现的。

- 当你向字典里添加一个元素时，需要同时指明这个元素的 key 和 value，在之后的操作中，你也可以修改这个值
- 字典常用于需要用 key 来读写 value 的情景 —— 这个操作摊还的时间复杂度被优化为 $O(1)$ ，亦即常数时间内可以完成。

[《Dive Into Python 3》2.7 Dictionaries](#) 描述了它的用法。

2.2 练习

请完成下述练习：

- [HackerRank. Hash Tables: Ransom Note](#)

3. 类

3.1 语法

3.1.1 定义类

[《Dive Into Python 3》7.2 Defining Classes](#) 描述了如何定义一个类。在此我们定义一个没有任何方法 (method) 和属性 (attribute) 的类：

```
class List:
    pass
```

3.1.2 类的实例化

```
l = List()
```

3.1.3 变量

[《Dive Into Python 3》7.4 Instance Variables](#) 描述了如何定义一个类的变量。

```
class List:
    def __init__(self, val, next):
        self.val = val
        self.next = next
```

3.2 练习

3.2.1 手动实现的链表 (非 Python 原生的 List)

请完成下述练习：

- 反向打印链表： [HackerRank. Print in Reverse](#)
 - （供参考，不用做）打印链表中的元素： [HackerRank. Print the Elements of a Linked List](#)
- 将节点插入到链表的某个位置上： [HackerRank. Insert a node at a specific position in a linked list.](#)
 - （供参考，不用做）子问题1: 在链表的尾部添加一个节点： [HackerRank. Insert a Node at the Tail of a Linked List](#)
 - （供参考，不用做）子问题2: 在链表的头部添加一个节点： [HackerRank. Insert a node at the head of a linked list](#)
- 删除节点： [HackerRank. Delete a Node](#)

3.2.2 树 (Tree)

用递归地方法完成下述两道练习：

- 树的中序遍历： [HackerRank. Tree: Inorder Traversal](#)
- 树的层次遍历： [HackerRank. Tree: Level Order Traversal](#)